

A g-2 Development Environment Tutorial

How to develop code using `gm2examples` and use the g-2 development environment.

This tutorial shows you how to set up your environment, set up a development area, check out a package (e.g. `gm2artexamples`), build it, run it, do some development, and share code.

1 Starting out fresh with a new development area

In this section, we will log into `gm2gpvm` and create a new development area.

If you do not have a `gm2gpvm` account, see the [New g-2 person welcome packet](#) for instructions for getting an account. See documentation on [GPCF](#) for how the machines work and how to configure your `ssh` on desktop/laptop.

Log in with `ssh gm2gpvm`. You will end up in your home area.

You can work in your home area, or utilize space on `/gm2/app/users`. If you do not have a directory there, you can make one easily with

```
mkdir /gm2/app/users/you
```

(Replace `you` with your user name). Go there with `cd /gm2/app/users/you`. This tutorial will assume you will use a directory in `/gm2/app/users`, but you can use your home area if you have room.

```
[lyon@gm2gpvm01 lyon]$ pwd
/gm2/app/users/lyon
[lyon@gm2gpvm01 lyon]$ cd /gm2/app/users/lyon
[lyon@gm2gpvm01 lyon]$ pwd
/gm2/app/users/lyon
```

Let's make a new directory for development. (I like to name development areas based on the date that I started).

```
[lyon@gm2gpvm01 work-20121003]$ mkdir work-20121003
[lyon@gm2gpvm01 lyon]$ cd work-20121003
```

Now, let's setup our development environment. You will always source the following script to start a **new** development area (if you want to continue development on an already existing development area, see a further section below).

```
[lyon@gm2gpvm01 work-20121003]$ source /gm2/app/software/prod/g-2/setup
g-2 software

--> To list gm2 releases, type
ups list -aK+ gm2

--> To use the latest gm2 release, type
setup gm2 v201209 -q e2:debug

--> After the set up above, use the "gm2d" command to do development stuff
```

You now must **setup** a g-2 release. The output typically tells you the latest release to use. You can get a list of all of the release by running the `ups` command in the output above.

```
[lyon@gm2gpvm01 work-20121003]$ ups list -aK+ gm2
"gm2" "v201209" "Linux64bit+2.6-2.5" "debug:e2" ""
"gm2" "v201209" "Linux64bit+2.6-2.5" "e2:prof" ""
```

So in this case, there are two g-2 releases. Both are version v201209. One is a debug build and the other is a “profile” build (we call optimized builds profile). This means that all of the dependent libraries are built with the appropriate compiler and linker flags. You should choose the appropriate build. Typically, you start with debug. That’s what we’ll do here. We’ll see later on how to switch to profile.

```
[lyon@gm2gpvm01 work-20121003]$ setup gm2 v201209 -q debug:e2
```

You can check what g-2 release you have set up with,

```
[lyon@gm2gpvm01 work-20121003]$ echo $GM2_VERSION
v201209
[lyon@gm2gpvm01 work-20121003]$ echo $GM2_FQ
slf5.x86_64.e2.debug
```

Setting up a g-2 release will give you access to the gm2d command, which makes some development processes easier.

```
[lyon@gm2gpvm01 work-20121003]$ gm2d
Usage gm2d (listTags | newDev | getRedmineGit | newProduct | setup_for_development | build | zapBuild | updateDepsCM | updateDepsPD) [-h for help]
gm2d -h gives more information
```

You can get more help with gm2d -h. We will use some of the commands in this tutorial. In general, you should learn and use the gm2d commands as they are (hopefully) easy to remember and will shield you from changes in the underlying infrastructure.

```
[lyon@gm2gpvm01 work-20121003]$ gm2d -h
Usage gm2d (listTags | newDev | getRedmineGit | newProduct | setup_for_development | build | zapBuild | updateDepsCM | updateDepsPD) [-h for help]"

Tools ( for help on tool, do "gm2d <tool> -h" )

newDev (n)           Start a new development area
getRedmineGit (g)    Clone a Redmine git repository
newProduct (p)       Create a new product from scratch
setup_for_development (s) Setup a development environment
build (b)            Run buildtool

zapBuild (z)         Delete everything in your build area
listTags (l)         List the git tags for a product
updateDepsCM (uc)    Update CMakeLists.txt file for latest dependencies
updateDepsPD (up)    Update product_deps file for latest dependencies
```

1.1 Initializing your new development area

Now that you’ve selected a g-2 release, you can initialize your development area for use. You only have to do this once for the development area. You can [learn more about the gm2d command](#), including command options that will not be discussed here.

```
[lyon@gm2gpvm01 work-20121003]$ gm2d newDev
NOTICE: Created srcs and build directories
NOTICE: Created srcs/CMakeLists.txt
NOTICE: Created srcs/setup_for_development
NOTICE: Created local products directory ./localProducts_v201209_debug-e2
NOTICE: Copied .upsfiles to ./localProducts_v201209_debug-e2
NOTICE: Created ./localProducts_v201209_debug-e2/setup

IMPORTANT: You must type
source ./localProducts_v201209_debug-e2/setup
NOW and whenever you log in
```

As the output states, you need run the source line to finish setting up your environment. This is the source line you will run when you want to continue development on this area if you log out and log back in later (see a section below).

```
[lyon@gm2gpvm01 work-20121003]$ source ./localProducts_v201209_debug-e2/setup
```

```
Executed setup gm2 v201209 -q debug:e2
```

The `gm2d newDev` command created three top level directories.

```
[lyon@gm2gpvm01 work-20121003]$ ls
build  localProducts_v201209_debug-e2  srcs
```

You can learn more about how the [development area is laid out](#). Here is a short summary:

- `srcs`: All of your source code lives within this directory.
- `build`: Your build “products” (files that are created by compiling and linking) go into the `build` directory.
- `localProducts...` This directory is tied to the particular g-2 environment and release you have set up. Within is a set up script you can use to restore your environment when you log back in again. If you want to do a practice release (not covered in this tutorial), those files will go here.

Let’s get some code to build. Let’s try the `gm2artexamples`. This code lives in our Redmine git repository. You can learn more about this code at <https://cdcv.fnl.gov/redmine/projects/gm2artexamples> (see especially the codedocify link under “Overview”). We need to `cd` to the `srcs` directory (remember, that’s where sources go) and execute `gm2d getRedmineGit`.

```
[lyon@gm2gpvm01 srcs]$ cd srcs
[lyon@gm2gpvm01 srcs]$ pwd
/gm2/app/users/lyon/work-20121003/srcs
[lyon@gm2gpvm01 srcs]$ gm2d getRedmineGit gm2artexamples
NOTICE: Running git clone ssh://p-gm2artexamples@cdcv.fnl.gov/cvs/projects/gm2artexamples
Cloning into 'gm2artexamples'...
warning: templates not found /gm2/app/home/gm2dat/makeGit/git-usr/share/git-core/templates
remote: Counting objects: 29  remote: Counting objects: 53  remote: Counting objects: 117  remote: Counting objects: 187  remote: Counting objects: 251  remote: Counting objects: 321
remote: Compressing objects: 0% (1/394)
remote: Compressing objects: 1% (4/394)
remote: Compressing objects: 2% (8/394)
remote: Compressing objects: 3% (12/394)
remote: Compressing objects: 4% (16/394)
remote: Compressing objects: 5% (20/394)
remote: Compressing objects: 6% (24/394)
remote: Compressing objects: 7% (28/394)
remote: Compressing objects: 8% (32/394)
remote: Compressing objects: 9% (36/394)  remote: Compressing objects: 10% (40/394)
remote: Compressing objects: 11% (44/394)
remote: Compressing objects: 12% (48/394)
remote: Compressing objects: 13% (52/394)
remote: Compressing objects: 14% (56/394)
remote: Compressing objects: 15% (60/394)
remote: Compressing objects: 16% (64/394)
remote: Compressing objects: 17% (67/394)
remote: Compressing objects: 18% (71/394)
remote: Compressing objects: 19% (75/394)
remote: Compressing objects: 20% (79/394)
remote: Compressing objects: 21% (83/394)
remote: Compressing objects: 22% (87/394)
remote: Compressing objects: 23% (91/394)
remote: Compressing objects: 24% (95/394)
remote: Compressing objects: 24% (96/394)
remote: Compressing objects: 25% (99/394)
remote: Compressing objects: 26% (103/394)
remote: Compressing objects: 27% (107/394)
remote: Compressing objects: 28% (111/394)
remote: Compressing objects: 29% (115/394)
remote: Compressing objects: 30% (119/394)
remote: Compressing objects: 31% (123/394)  remote: Compressing objects: 32% (127/394)
remote: Compressing objects: 33% (131/394)
remote: Compressing objects: 34% (134/394)
remote: Compressing objects: 35% (138/394)
remote: Compressing objects: 36% (142/394)
remote: Compressing objects: 37% (146/394)
```

```
remote: Compressing objects: 38% (150/394)
remote: Compressing objects: 39% (154/394)
remote: Compressing objects: 40% (158/394)
remote: Compressing objects: 41% (162/394)
remote: Compressing objects: 42% (166/394)
remote: Compressing objects: 43% (170/394)
remote: Compressing objects: 44% (174/394)
remote: Compressing objects: 45% (178/394)
remote: Compressing objects: 46% (182/394)
remote: Compressing objects: 46% (183/394)
remote: Compressing objects: 47% (186/394)
remote: Compressing objects: 48% (190/394)
remote: Compressing objects: 49% (194/394)
remote: Compressing objects: 50% (197/394)
remote: Compressing objects: 51% (201/394)
remote: Compressing objects: 52% (205/394)
remote: Compressing objects: 53% (209/394) remote: Compressing objects: 54% (213/394)
remote: Compressing objects: 55% (217/394)
remote: Compressing objects: 56% (221/394)
remote: Compressing objects: 57% (225/394)
remote: Compressing objects: 58% (229/394)
remote: Compressing objects: 59% (233/394)
remote: Compressing objects: 60% (237/394)
remote: Compressing objects: 61% (241/394)
remote: Compressing objects: 62% (245/394)
remote: Compressing objects: 63% (249/394)
remote: Compressing objects: 64% (253/394)
remote: Compressing objects: 65% (257/394)
remote: Compressing objects: 66% (261/394)
remote: Compressing objects: 67% (264/394)
remote: Compressing objects: 68% (268/394)
remote: Compressing objects: 69% (272/394)
remote: Compressing objects: 70% (276/394)
remote: Compressing objects: 71% (280/394)
remote: Compressing objects: 72% (284/394)
remote: Compressing objects: 73% (288/394)
remote: Compressing objects: 74% (292/394)
remote: Compressing objects: 75% (296/394)
remote: Compressing objects: 76% (300/394)
remote: Compressing objects: 76% (303/394)
remote: Compressing objects: 77% (304/394)
remote: Compressing objects: 78% (308/394)
remote: Compressing objects: 79% (312/394)
remote: Compressing objects: 80% (316/394)
remote: Compressing objects: 81% (320/394)
remote: Compressing objects: 82% (324/394)
remote: Compressing objects: 83% (328/394)
remote: Compressing objects: 84% (331/394)
remote: Compressing objects: 85% (335/394)
remote: Compressing objects: 86% (339/394)
remote: Compressing objects: 87% (343/394)
remote: Compressing objects: 88% (347/394)
remote: Compressing objects: 89% (351/394) remote: Compressing objects: 90% (355/394)
remote: Compressing objects: 91% (359/394)
remote: Compressing objects: 92% (363/394)
remote: Compressing objects: 93% (367/394)
remote: Compressing objects: 94% (371/394)
remote: Compressing objects: 95% (375/394)
remote: Compressing objects: 96% (379/394)
remote: Compressing objects: 97% (383/394)
remote: Compressing objects: 98% (387/394)
remote: Compressing objects: 99% (391/394)
remote: Compressing objects: 100% (394/394)
remote: Compressing objects: 100% (394/394), done.
Receiving objects: 0% (1/401)
Receiving objects: 1% (5/401)
Receiving objects: 2% (9/401)
```

```
Receiving objects: 3% (13/401)
Receiving objects: 4% (17/401)
Receiving objects: 5% (21/401)
Receiving objects: 6% (25/401)
Receiving objects: 7% (29/401)
Receiving objects: 8% (33/401)
Receiving objects: 9% (37/401)
Receiving objects: 10% (41/401)
Receiving objects: 11% (45/401)
Receiving objects: 12% (49/401)
Receiving objects: 13% (53/401)
Receiving objects: 14% (57/401)
Receiving objects: 15% (61/401)
Receiving objects: 16% (65/401)
Receiving objects: 17% (69/401)
Receiving objects: 18% (73/401)
Receiving objects: 19% (77/401)
Receiving objects: 20% (81/401)
Receiving objects: 21% (85/401)
Receiving objects: 22% (89/401)
Receiving objects: 23% (93/401)
Receiving objects: 24% (97/401)
Receiving objects: 25% (101/401)
Receiving objects: 26% (105/401)
Receiving objects: 27% (109/401)
Receiving objects: 28% (113/401)
Receiving objects: 29% (117/401)
Receiving objects: 30% (121/401)
Receiving objects: 31% (125/401)
Receiving objects: 32% (129/401)
Receiving objects: 33% (133/401)
Receiving objects: 34% (137/401)
Receiving objects: 35% (141/401)
Receiving objects: 36% (145/401)
Receiving objects: 37% (149/401)
Receiving objects: 38% (153/401)
Receiving objects: 39% (157/401)
Receiving objects: 40% (161/401)
Receiving objects: 41% (165/401)
Receiving objects: 42% (169/401)
Receiving objects: 43% (173/401)
Receiving objects: 44% (177/401)
Receiving objects: 45% (181/401)
Receiving objects: 46% (185/401)
Receiving objects: 47% (189/401)
Receiving objects: 48% (193/401)
Receiving objects: 49% (197/401)
Receiving objects: 50% (201/401)
Receiving objects: 51% (205/401)
Receiving objects: 52% (209/401)
Receiving objects: 53% (213/401)
Receiving objects: 54% (217/401)
Receiving objects: 55% (221/401), 76.00 KiB | 127 KiB/s
Receiving objects: 56% (225/401), 76.00 KiB | 127 KiB/s
Receiving objects: 57% (229/401), 76.00 KiB | 127 KiB/s
Receiving objects: 58% (233/401), 76.00 KiB | 127 KiB/s
Receiving objects: 59% (237/401), 76.00 KiB | 127 KiB/s
Receiving objects: 60% (241/401), 76.00 KiB | 127 KiB/s
Receiving objects: 61% (245/401), 76.00 KiB | 127 KiB/s
Receiving objects: 62% (249/401), 76.00 KiB | 127 KiB/s
Receiving objects: 63% (253/401), 76.00 KiB | 127 KiB/s
Receiving objects: 64% (257/401), 76.00 KiB | 127 KiB/s
Receiving objects: 65% (261/401), 76.00 KiB | 127 KiB/s
Receiving objects: 66% (265/401), 76.00 KiB | 127 KiB/s
Receiving objects: 67% (269/401), 76.00 KiB | 127 KiB/s
Receiving objects: 68% (273/401), 76.00 KiB | 127 KiB/s
Receiving objects: 69% (277/401), 76.00 KiB | 127 KiB/s
Receiving objects: 70% (281/401), 76.00 KiB | 127 KiB/s
```

```
Receiving objects: 71% (285/401), 76.00 KiB | 127 KiB/s
Receiving objects: 72% (289/401), 76.00 KiB | 127 KiB/s
Receiving objects: 73% (293/401), 76.00 KiB | 127 KiB/s
Receiving objects: 74% (297/401), 76.00 KiB | 127 KiB/s
Receiving objects: 75% (301/401), 76.00 KiB | 127 KiB/s
Receiving objects: 76% (305/401), 76.00 KiB | 127 KiB/s
Receiving objects: 77% (309/401), 76.00 KiB | 127 KiB/s
Receiving objects: 78% (313/401), 76.00 KiB | 127 KiB/s
Receiving objects: 79% (317/401), 76.00 KiB | 127 KiB/s
Receiving objects: 80% (321/401), 76.00 KiB | 127 KiB/s
Receiving objects: 81% (325/401), 76.00 KiB | 127 KiB/s
Receiving objects: 82% (329/401), 76.00 KiB | 127 KiB/s
Receiving objects: 83% (333/401), 76.00 KiB | 127 KiB/s
Receiving objects: 84% (337/401), 76.00 KiB | 127 KiB/s
Receiving objects: 85% (341/401), 76.00 KiB | 127 KiB/s
Receiving objects: 86% (345/401), 76.00 KiB | 127 KiB/s
Receiving objects: 87% (349/401), 76.00 KiB | 127 KiB/s
Receiving objects: 88% (353/401), 76.00 KiB | 127 KiB/s
Receiving objects: 89% (357/401), 76.00 KiB | 127 KiB/s
Receiving objects: 90% (361/401), 76.00 KiB | 127 KiB/s
Receiving objects: 91% (365/401), 76.00 KiB | 127 KiB/s
Receiving objects: 92% (369/401), 76.00 KiB | 127 KiB/s
Receiving objects: 93% (373/401), 76.00 KiB | 127 KiB/s
Receiving objects: 94% (377/401), 76.00 KiB | 127 KiB/s
Receiving objects: 95% (381/401), 76.00 KiB | 127 KiB/s
Receiving objects: 96% (385/401), 76.00 KiB | 127 KiB/s
Receiving objects: 97% (389/401), 76.00 KiB | 127 KiB/s
remote: Total 401 (delta 240), reused 0 (delta 0)
Receiving objects: 98% (393/401), 76.00 KiB | 127 KiB/s
Receiving objects: 99% (397/401), 76.00 KiB | 127 KiB/s
Receiving objects: 100% (401/401), 76.00 KiB | 127 KiB/s
Receiving objects: 100% (401/401), 104.95 KiB | 127 KiB/s, done.
Resolving deltas: 0% (0/240)
Resolving deltas: 1% (3/240)
Resolving deltas: 4% (11/240)
Resolving deltas: 5% (12/240)
Resolving deltas: 7% (18/240)
Resolving deltas: 8% (21/240)
Resolving deltas: 10% (24/240)
Resolving deltas: 11% (28/240)
Resolving deltas: 12% (29/240)
Resolving deltas: 13% (33/240)
Resolving deltas: 15% (37/240)
Resolving deltas: 16% (39/240)
Resolving deltas: 17% (42/240)
Resolving deltas: 19% (47/240)
Resolving deltas: 21% (51/240)
Resolving deltas: 25% (61/240)
Resolving deltas: 27% (65/240)
Resolving deltas: 29% (70/240)
Resolving deltas: 32% (78/240)
Resolving deltas: 34% (83/240)
Resolving deltas: 35% (86/240)
Resolving deltas: 36% (87/240)
Resolving deltas: 38% (93/240)
Resolving deltas: 40% (97/240)
Resolving deltas: 41% (99/240)
Resolving deltas: 42% (102/240)
Resolving deltas: 43% (105/240)
Resolving deltas: 44% (106/240)
Resolving deltas: 45% (108/240)
Resolving deltas: 47% (113/240)
Resolving deltas: 54% (131/240)
Resolving deltas: 57% (139/240)
Resolving deltas: 61% (147/240)
Resolving deltas: 67% (162/240)
Resolving deltas: 70% (168/240)
```

```

Resolving deltas: 78% (188/240)
Resolving deltas: 81% (195/240)
Resolving deltas: 82% (198/240)
Resolving deltas: 84% (202/240)
Resolving deltas: 87% (209/240)
Resolving deltas: 88% (212/240)
Resolving deltas: 90% (216/240)
Resolving deltas: 91% (219/240)
Resolving deltas: 92% (221/240)
Resolving deltas: 93% (224/240)
Resolving deltas: 95% (229/240)
Resolving deltas: 96% (232/240)
Resolving deltas: 97% (235/240)
Resolving deltas: 98% (236/240)
Resolving deltas: 99% (238/240)
Resolving deltas: 100% (240/240)
Resolving deltas: 100% (240/240), done.
NOTICE: Added gm2artexamples to CMakeLists.txt file
Using default branch names.
NOTICE: Making develop a tracking branch of origin/develop
Switched to branch 'master'
Deleted branch develop (was 56bd129).
Branch develop set up to track remote branch develop from origin.
Switched to branch 'develop'
NOTICE: You can now 'cd gm2artexamples'

You are now on the develop branch (check with 'git branch')
To make a new feature, do 'git flow feature start <featureName>'

```

The capture of the output above looks a bit messy due to progress bars being displayed. The output on your screen will look much cleaner. There are several steps that the `gm2d getRedmineGit` command does for you:

- Performs a `git clone` on the repository in Redmine (it also figures out the git ssh “URL” so you don’t have to). This step clones the repository to your directory.
- Switches you to the “develop” branch and sets it up for tracking more on that below. This is the branch we’ll use for the shared development code from others. You will, however, use a **feature branch** for your development.
- Initialize `git flow`. `git flow` is a nice add-on to git that makes creating feature branches and merging them back into develop easy.

You should never develop directly on the master branch. Instead, make a feature branch. Once you have finished developing your feature branch, you will merge that code back into `develop`.

1.2 Creating a feature branch

Creating a feature branch is easy.

```

[lyon@gm2gpvm01 gm2artexamples (feature/myNewFeature)]$ cd gm2artexamples
[lyon@gm2gpvm01 gm2artexamples (develop=)]$ git flow feature start myNewFeature
Switched to a new branch 'feature/myNewFeature'

```

```

Summary of actions:
- A new branch 'feature/myNewFeature' was created, based on 'develop'
- You are now on branch 'feature/myNewFeature'

```

Now, start committing on your feature. When done, use:

```
git flow feature finish myNewFeature
```

You can always check what branch you are on with

```

[lyon@gm2gpvm01 gm2artexamples (feature/myNewFeature)]$ git branch
develop
* feature/myNewFeature
master

```

Here are listed all of the branches in **your** repository (later, we'll see what branches are on Redmine). The branch you have currently checked out is marked with the star. Notice how `feature/` is prepended to the actual name of the branch. When using `git flow`, you do not need to specify `feature/`.

1.3 Prepare to build code

Before you can build code, you need to set up the environment for the application you are going to build (e.g. perhaps it depends on Art, or Root or Geant). A script called `setup_for_development` exists to prepare this environment. You need to run this script before you can build. Once it is run, you don't need to run it again unless you log back out and back in again, or you drastically change the build (e.g. build a different or an additional application in the same development area, or change the build type from debug to optimized). Your current directory must be the build directory. The build system will remember this directory and will direct the build files to go there regardless of where you launch the build from.

You can use the command `source gm2d setup_for_development -buildType`

Note that you must `source` the command since it changes environment variables. There are some short cuts. You can replace `setup_for_development` with `s`.

The `-buildType` is `-d` for a debug build, `-p` for a profile build (see below), or `-n` for a non-compile build (not covered in this tutorial). The build type must match the g-2 release that you selected earlier (e.g. debug, profile, etc).

```
[lyon@gm2gpvm01 gm2artexamples (feature/myNewFeature)]$ cd ../../build
[lyon@gm2gpvm01 build]$ pwd
/gm2/app/users/lyon/work-20121003/build
[lyon@gm2gpvm01 build]$ source gm2d setup_for_development -d
--- Running gm2artexamples setup_for_development
The working build directory is /gm2/app/users/lyon/work-20121003/build
The source code directory is /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples
----- check this block for errors -----
-----
/gm2/app/users/lyon/work-20121003/build/lib has been added to LD_LIBRARY_PATH
/gm2/app/users/lyon/work-20121003/build/bin has been added to PATH

CETPKG_SOURCE=/gm2/app/users/lyon/work-20121003/srcs/gm2artexamples
CETPKG_BUILD=/gm2/app/users/lyon/work-20121003/build
CETPKG_NAME=gm2artexamples
CETPKG_VERSION=v0_3
CETPKG_QUAL=e2:debug
CETPKG_TYPE=Debug

Please use this cmake command:
cmake -DCMAKE_INSTALL_PREFIX=/install/path -DCMAKE_BUILD_TYPE=$CETPKG_TYPE $CETPKG_SOURCE
Setting GM2ARTEXAMPLES_DIR=/gm2/app/users/lyon/work-20121003/build/gm2artexamples
-----
Define LD_LIBRARY_PATH to include /gm2/app/users/lyon/work-20121003/build/lib
Define FHICL_FILE_PATH to include /gm2/app/users/lyon/work-20121003/build/fcl
Summary:
  CETPKG_SOURCE   = /gm2/app/users/lyon/work-20121003/srcs
  CETPKG_BUILD    = /gm2/app/users/lyon/work-20121003/build
  CETPKG_QUAL     = e2:debug
  CETPKG_TYPE     = CETPKG_TYPE
  CETPKG_INSTALL = /gm2/app/users/lyon/work-20121003/localProducts_v201209_debug-e2
Run buildtool to build
```

There is lots of output from the command. It is important to look between the dashed lines where it says "Check this block for errors". There are no errors here. If you see something like,

```
ERROR: Product 'messagefacility' (with qualifiers 'a7:debug'), has no v1_10_08 version (or may not exist)
ERROR: setup of required products has failed
```

then you have an **inconsistent** build environment and your build will not succeed. What's happening here is that your application demands dependent products of a certain version and build type (e.g. debug). If those versions do not exist or you have other versions already set up, then there is a problem. The problem could be,

- The application dependencies are old and need to be updated for this g-2 release (that's the case here).
- Your application is demanding a product of a different version than the g-2 release already has set up. The application dependencies must be updated, or you have the wrong release of g-2 set up.

- You have a typo in the dependency specification.

All of these problems are beyond the scope of this tutorial. Seeing such errors should be rare.

Finally, the command says to run `buildtool` to build. This will change in a future release. Instead, to build we will use another `gm2d` command.

1.4 Building the code

To build the code, we will use the `gm2d build` command. You can also do `gm2d b` for short.

```
[lyon@gm2gpvm01 build]$ gm2d build
INFO: CETPKG_TYPE = CETPKG_TYPE

-----
INFO: Stage cmake.
-----

-- The C compiler identification is GNU 4.7.1
-- The CXX compiler identification is GNU 4.7.1
-- Check for working C compiler: /gm2/app/software/prod/external/gcc/v4_7_1/Linux64bit+2.6-2.5/bin/gcc
-- Check for working C compiler: /gm2/app/software/prod/external/gcc/v4_7_1/Linux64bit+2.6-2.5/bin/gcc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /gm2/app/software/prod/external/gcc/v4_7_1/Linux64bit+2.6-2.5/bin/c++
-- Check for working CXX compiler: /gm2/app/software/prod/external/gcc/v4_7_1/Linux64bit+2.6-2.5/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Product is gm2artexamples v0_3 e2:debug
-- Module path is /gm2/app/software/prod/g-2/gm2softwaretools/v201209_01/Modules;/gm2/app/software/prod/external/cetbuildtools/v2_03_03/Modules
-- version parses to v0_3.v0_3.v0_3
-- project version is v0_3.v0_3.v0_3
-- Building for Linux slf5 x86_64
-- set_flavor_qual: flavorqual directory is gm2artexamples/v0_3/slf5.x86_64.e2.debug
-- Selected diagnostics option VIGILANT
-- cmake build type set to CETPKG_TYPE in directory gm2artexamples and below
-- compiler flags for directory gm2artexamples and below
--   C++ FLAGS:      -std=c++0x -Wno-deprecated-declarations
--   C   FLAGS:
-- CETBUILDTOOLS version and qualifier are v2_03_03
-- CPP0X version and qualifier are v1_03_14 debug:e2
-- LIBSIGCPP version and qualifier are v2_2_10 debug:e2
-- CPPUNIT version and qualifier are v1_12_1 debug:e2
-- CLHEP version and qualifier are v2_1_2_4 debug:e2
-- CETLIB version and qualifier are v1_03_14 debug:e2
-- GCCXML version and qualifier are v0_9_20120702 gcc47
-- ART version and qualifier are v1_02_02 debug:e2
-- FHICLCP version and qualifier are v2_17_01 debug:e2
-- MESSAGEFACILITY version and qualifier are v1_10_15 debug:e2
-- ROOT version and qualifier are v5_34_01 debug:e2
-- Boost version and qualifier are v1_50_0 debug:e2
-- Boost version: 1.50.0
-- Found the following Boost libraries:
--   filesystem
--   program_options
--   regex
--   thread
--   unit_test_framework
-- art_make: no library for /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/Lesson1
-- art_make: no library for /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/Lesson2
-- full_qualifiers: e2:debug e2-debug
-- CPACK_PACKAGE_NAME and CPACK_SYSTEM_NAME are gm2artexamples slf5-x86_64-gcc47-e2-debug
```

```
-- Configuring done
-- Generating done
-- Build files have been written to: /gm2/app/users/lyon/work-20121003/build
```

```
-----
INFO: Stage cmake successful.
-----
```

```
-----
INFO: gm2artexamples version v0_3.v0_3 configured.
-----
```

```
-----
INFO: Stage build.
-----
```

```
Scanning dependencies of target gm2artexamples_DataObjects
[ 5%] Building CXX object gm2artexamples/DataObjects/CMakeFiles/gm2artexamples_DataObjects.dir/MyLittleDatum.cc.o
Linking CXX shared library ../lib/libgm2artexamples_DataObjects.so
[ 5%] Built target gm2artexamples_DataObjects
[ 10%] Generating gm2artexamples_DataObjects_dict.cpp, gm2artexamples_DataObjects_map.cpp
---> genreflex: INFO: Using gccxml from /gm2/app/software/prod/external/gccxml/v0_9_20120702/Linux64bit+2.6-2.5-gcc47/bin/gccxml
---> genreflex: INFO: Parsing file /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/DataObjects/classes.h with GCC_XML OK
---> genreflex: INFO: Generating Reflex Dictionary
class artex::MyLittleDatum
class std::vector<artex::MyLittleDatum>
class art::Wrapper<std::vector<artex::MyLittleDatum> >
Scanning dependencies of target gm2artexamples_DataObjects_dict
[ 15%] Building CXX object gm2artexamples/DataObjects/CMakeFiles/gm2artexamples_DataObjects_dict.dir/gm2artexamples_DataObjects_dict.cpp.o
Linking CXX shared library ../lib/libgm2artexamples_DataObjects_dict.so
[ 15%] Built target gm2artexamples_DataObjects_dict
Scanning dependencies of target gm2artexamples_DataObjects_map
[ 20%] Building CXX object gm2artexamples/DataObjects/CMakeFiles/gm2artexamples_DataObjects_map.dir/gm2artexamples_DataObjects_map.cpp.o
Linking CXX shared library ../lib/libgm2artexamples_DataObjects_map.so
[ 25%] Built target gm2artexamples_DataObjects_map
[ 30%] Generating gm2artexamples_HitAndTrackObjects_dict.cpp, gm2artexamples_HitAndTrackObjects_map.cpp
---> genreflex: INFO: Using gccxml from /gm2/app/software/prod/external/gccxml/v0_9_20120702/Linux64bit+2.6-2.5-gcc47/bin/gccxml
---> genreflex: INFO: Parsing file /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/HitAndTrackObjects/classes.h with GCC_XML OK
---> genreflex: INFO: Generating Reflex Dictionary
class artex::Hit
class std::vector<artex::Hit>
class CLHEP::Hep3Vector
class art::Wrapper<std::vector<artex::Hit> >
Scanning dependencies of target gm2artexamples_HitAndTrackObjects_dict
[ 35%] Building CXX object gm2artexamples/HitAndTrackObjects/CMakeFiles/gm2artexamples_HitAndTrackObjects_dict.dir/gm2artexamples_HitAndTrackObjects_dict.cpp.o
Linking CXX shared library ../lib/libgm2artexamples_HitAndTrackObjects_dict.so
[ 35%] Built target gm2artexamples_HitAndTrackObjects_dict
Scanning dependencies of target gm2artexamples_HitAndTrackObjects_map
[ 40%] Building CXX object gm2artexamples/HitAndTrackObjects/CMakeFiles/gm2artexamples_HitAndTrackObjects_map.dir/gm2artexamples_HitAndTrackObjects_map.cpp.o
Linking CXX shared library ../lib/libgm2artexamples_HitAndTrackObjects_map.so
[ 45%] Built target gm2artexamples_HitAndTrackObjects_map
Scanning dependencies of target gm2artexamples_hitandtrackobjects
[ 50%] Building CXX object gm2artexamples/HitAndTrackObjects/CMakeFiles/gm2artexamples_hitandtrackobjects.dir/Hit.cpp.o
Linking CXX shared library ../lib/libgm2artexamples_hitandtrackobjects.so
[ 50%] Built target gm2artexamples_hitandtrackobjects
Scanning dependencies of target gm2artexamples_Lesson1_HelloWorld1_module
[ 55%] Building CXX object gm2artexamples/Lesson1/CMakeFiles/gm2artexamples_Lesson1_HelloWorld1_module.dir/HelloWorld1_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson1_HelloWorld1_module.so
[ 55%] Built target gm2artexamples_Lesson1_HelloWorld1_module
Scanning dependencies of target gm2artexamples_Lesson1_HelloWorld2_module
[ 60%] Building CXX object gm2artexamples/Lesson1/CMakeFiles/gm2artexamples_Lesson1_HelloWorld2_module.dir/HelloWorld2_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson1_HelloWorld2_module.so
[ 60%] Built target gm2artexamples_Lesson1_HelloWorld2_module
Scanning dependencies of target gm2artexamples_Lesson1_MyDatumReader_module
[ 65%] Building CXX object gm2artexamples/Lesson1/CMakeFiles/gm2artexamples_Lesson1_MyDatumReader_module.dir/MyDatumReader_module.cc.o
```

```

Linking CXX shared library ../lib/libgm2artexamples_Lesson1_MyDatumReader_module.so
[ 65%] Built target gm2artexamples_Lesson1_MyDatumReader_module
Scanning dependencies of target gm2artexamples_Lesson1_ProduceMyLittleDatum_module
[ 70%] Building CXX object gm2artexamples/Lesson1/CMakeFiles/gm2artexamples_Lesson1_ProduceMyLittleDatum_module.dir/ProduceMyLittleDatum_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson1_ProduceMyLittleDatum_module.so
[ 70%] Built target gm2artexamples_Lesson1_ProduceMyLittleDatum_module
Scanning dependencies of target gm2artexamples_Lesson2_makeHits_module
[ 75%] Building CXX object gm2artexamples/Lesson2/CMakeFiles/gm2artexamples_Lesson2_makeHits_module.dir/makeHits_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson2_makeHits_module.so
[ 75%] Built target gm2artexamples_Lesson2_makeHits_module
Scanning dependencies of target gm2artexamples_Lesson2_readHits_module
[ 80%] Building CXX object gm2artexamples/Lesson2/CMakeFiles/gm2artexamples_Lesson2_readHits_module.dir/readHits_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson2_readHits_module.so
[ 80%] Built target gm2artexamples_Lesson2_readHits_module
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!MyLittleDatum_test.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!MyLittleDatum_test.d
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!MyLittleDatum_test.d!fcl!MyLittleDatum_test.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!MyLittleDatum_test.d!fcl!MyLittleDatum_test.fcl
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!MyLittleDatum_test.d!fcl!messageDefaults.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!MyLittleDatum_test.d!fcl!messageDefaults.fcl
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!MyLittleDatum_r.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!MyLittleDatum_r.fcl
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!MyLittleDatum_w.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!MyLittleDatum_w.fcl
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!messageDefaults.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!messageDefaults.fcl
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!simple_test.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!simple_test.d
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!test_with_boost.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!test_with_boost.d
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!very_simple_test.sh.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!very_simple_test.sh.d
Scanning dependencies of target gm2artexamples_test_MyLittleDatumAnalyzer_module
[ 85%] Building CXX object gm2artexamples/test/CMakeFiles/gm2artexamples_test_MyLittleDatumAnalyzer_module.dir/MyLittleDatumAnalyzer_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_test_MyLittleDatumAnalyzer_module.so
[ 85%] Built target gm2artexamples_test_MyLittleDatumAnalyzer_module
Scanning dependencies of target gm2artexamples_test_MyLittleDatumProducer_module
[ 90%] Building CXX object gm2artexamples/test/CMakeFiles/gm2artexamples_test_MyLittleDatumProducer_module.dir/MyLittleDatumProducer_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_test_MyLittleDatumProducer_module.so
[ 90%] Built target gm2artexamples_test_MyLittleDatumProducer_module
Scanning dependencies of target myLittleDatum_wr.sh
[ 90%] Built target myLittleDatum_wr.sh
Scanning dependencies of target simple_test
[ 95%] Building CXX object gm2artexamples/test/CMakeFiles/simple_test.dir/simple_test.cc.o
Linking CXX executable ../bin/simple_test
[ 95%] Built target simple_test
Scanning dependencies of target test_with_boost
[100%] Building CXX object gm2artexamples/test/CMakeFiles/test_with_boost.dir/test_with_boost.cc.o
Linking CXX executable ../bin/test_with_boost
[100%] Built target test_with_boost
Scanning dependencies of target very_simple_test.sh
[100%] Built target very_simple_test.sh

real    3m23.502s
user    1m26.817s
sys      0m14.188s

-----
INFO: Stage build successful.
-----

```

Again, the output on your screen will be cleaner. We are apparently left with a good build! Let's run the program. In this case, gm2artexamples builds some Art examples. Let's try one.

1.5 Running Art

The command to run Art is gm2 (nice!). Though be careful not to confuse it with gm2d. We need a FHICL configuration file. Look in the fcl directory.

```
[lyon@gm2gpvm01 fcl]$ ls fcl
hello1.fcl  makeAndReadDatum.fcl  makeHits.fcl      minimalMessageService.fcl  readHits.fcl
hello2.fcl  makeDatum.fcl           messageservice.fcl  readDatum.fcl
```

Let's run hello1.fcl. Note that we do not have to change directory.

```
[lyon@gm2gpvm01 build]$ gm2 -c hello1.fcl
%MSG-i MF_INIT_OK:  gm2 04-Oct-2012 00:29:31 CDT JobSetup
MessageLogger initialization complete.
%MSG
Begin processing the 1st record. run: 1 subRun: 0 event: 1 at 04-Oct-2012 00:29:33 CDT
Hello, world. From analyze. run: 1 subRun: 0 event: 1
Begin processing the 2nd record. run: 1 subRun: 0 event: 2 at 04-Oct-2012 00:29:33 CDT
Hello, world. From analyze. run: 1 subRun: 0 event: 2

TrigReport ----- Event Summary -----
TrigReport Events total = 2 passed = 2 failed = 0

TrigReport ----- Modules in End-Path: path1 -----
TrigReport Trig Bit# Visited Passed Failed Error Name
TrigReport 0 0 2 2 0 0 hello

TimeReport ----- Time Summary ---[sec]----
TimeReport CPU = 0.000000 Real = 0.002000

Art has completed and will exit with status 0.
```

It worked!

2 Restoring your environment when logging in

Let's log out of the gm2gpvm machine. Now when we log back in later, we want to continue our development in our same development area. We must restore the environment we had before. That involves doing two things:

1. Choosing the same g-2 release as before
2. Running setup_for_development

These two steps restore your environment and allow you to continue development.

Here is how to perform those steps after logging in,

```
[lyon@gm2gpvm01 work-20121003]$ pwd # We're in our home area
/gm2/app/users/lyon/work-20121003
[lyon@gm2gpvm01 work-20121003]$ cd /gm2/app/users/lyon/work-20121003 # Go to our development area
[lyon@gm2gpvm01 work-20121003]$ pwd
/gm2/app/users/lyon/work-20121003
[lyon@gm2gpvm01 work-20121003]$ ls
build localProducts_v201209_debug-e2 srcs
```

The script localProducts_v201209_debug/setup a convenient way to restore the g-2 release,

```
[lyon@gm2gpvm01 work-20121003]$ source localProducts_v201209_debug-e2/setup
Executed setup gm2 v201209 -q debug:e2
```

That's done. Now we have to restore the application specific environment variables with `setup_for_development`.

```
[lyon@gm2gpvm01 build]$ ccd build # Always be in the build directory
[lyon@gm2gpvm01 build]$ source gm2d s -d
--- Running gm2artexamples setup_for_development
The working build directory is /gm2/app/users/lyon/work-20121003/build
The source code directory is /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples
----- check this block for errors -----
-----
/gm2/app/users/lyon/work-20121003/build/lib has been added to LD_LIBRARY_PATH
/gm2/app/users/lyon/work-20121003/build/bin has been added to PATH

CETPKG_SOURCE=/gm2/app/users/lyon/work-20121003/srcs/gm2artexamples
CETPKG_BUILD=/gm2/app/users/lyon/work-20121003/build
CETPKG_NAME=gm2artexamples
CETPKG_VERSION=v0_3
CETPKG_QUAL=e2:debug
CETPKG_TYPE=Debug

Please use this cmake command:
cmake -DCMAKE_INSTALL_PREFIX=/install/path -DCMAKE_BUILD_TYPE=$CETPKG_TYPE $CETPKG_SOURCE
    Setting GM2ARTEEXAMPLES_DIR=/gm2/app/users/lyon/work-20121003/build/gm2artexamples
-----
Define LD_LIBRARY_PATH to include /gm2/app/users/lyon/work-20121003/build/lib
Define FHICL_FILE_PATH to include /gm2/app/users/lyon/work-20121003/build/fcl
Summary:
  CETPKG_SOURCE   = /gm2/app/users/lyon/work-20121003/srcs
  CETPKG_BUILD    = /gm2/app/users/lyon/work-20121003/build
  CETPKG_QUAL     = e2:debug
  CETPKG_TYPE     = CETPKG_TYPE
  CETPKG_INSTALL  = /gm2/app/users/lyon/work-20121003/localProducts_v201209_debug-e2
Run buildtool to build
```

We're back in business. You can now run, or build and run.

3 Changing and committing code

Let's make a minor change to the code. Instead of example `hello1` printing "Hello, World" it should read "Hello, g-2".

Ordinarily, you would run some editor to make this change. We'll do it with "sed".

First, we need to change directory to the source code. The `setup` script in the `localProducts` directory defines a convenient variable `$SRCS` which has the path to your `srcs` directory.

```
[lyon@gm2gpvm01 build]$ pwd
/gm2/app/users/lyon/work-20121003/build
[lyon@gm2gpvm01 build]$ echo $SRCS
/gm2/app/users/lyon/work-20121003/srcs
[lyon@gm2gpvm01 build]$ cd $SRCS
```

Let's see what's here.

```
[lyon@gm2gpvm01 srcs]$ ls
CMakeLists.txt  gm2artexamples  setup_for_development
```

There are two files, CMakeLists.txt and setup_for_development, and the gm2artexamples source code directory. The two files are used by the build system and are not typically edited by users. Let's find our hello code.

```
[lyon@gm2gpvm01 srcs]$ cd gm2artexamples
[lyon@gm2gpvm01 gm2artexamples (feature/myNewFeature)]$ ls
CMakeLists.txt  DataObjects  fcl  HitAndTrackObjects  Lesson1  Lesson2  readme.doonly  readme.html  test  ups
[lyon@gm2gpvm01 gm2artexamples (feature/myNewFeature)]$ cd Lesson1
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ ls
CMakeLists.txt  HelloWorld1_module.cc  HelloWorld2_module.cc  MyDatumReader_module.cc  ProduceMyLittleDatum_module.cc
```

Ahh – it's in HelloWorld1_module.cc.

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ grep -i world HelloWorld1_module.cc
// The @HelloWorld1@ ART module or plugin
// <pre>artmod analyzer artex::HelloWorld1</pre>
// Declare the class for our analyzer, @HelloWorld1@. Note that it inherits
class HelloWorld1 : public art::EDAnalyzer {
    explicit HelloWorld1(fhicl::ParameterSet const& pset);
    virtual ~HelloWorld1();
    HelloWorld1::HelloWorld1(fhicl::ParameterSet const& ) {
        HelloWorld1::~HelloWorld1() {}
    void HelloWorld1::analyze(const art::Event& event){
        mf::LogVerbatim("test") << "Hello, world. From analyze. " << event.id();
using artex::HelloWorld1;
DEFINE_ART_MODULE(HelloWorld1)
```

The mf::LogVerbatim line is the one we want to change.

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ sed -i 's/, world/, g-2/' HelloWorld1_module.cc
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature *)]$ grep g-2 HelloWorld1_module.cc
// (see "here":https://cdcv.sfnal.gov/redmine/projects/g-2/wiki/Artmod for information about @artmod@ )
mf::LogVerbatim("test") << "Hello, g-2. From analyze. " << event.id();
```

We've made our change. Let's build again. Note that we don't have to change directories.

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature *)]$ gm2d b
INFO: CETPKG_TYPE = CETPKG_TYPE

-----
INFO: Stage cmake.
-----

-- Product is gm2artexamples v0_3 e2:debug
-- Module path is /gm2/app/software/prod/g-2/gm2softwaretools/v201209_01/Modules;/gm2/app/software/prod/external/cetbuildtools/v2_03_03/Modules
-- version parses to v0_3.v0_3.v0_3
-- project version is v0_3.v0_3.v0_3
-- Building for Linux slf5 x86_64
-- set_flavor_qual: flavorqual directory is gm2artexamples/v0_3/slf5.x86_64.e2.debug
-- Selected diagnostics option VIGILANT
-- cmake build type set to CETPKG_TYPE in directory gm2artexamples and below
-- compiler flags for directory gm2artexamples and below
-- C++ FLAGS: -std=c++0x -Wno-deprecated-declarations
-- C FLAGS:
-- CETBUILDTOOLS version and qualifier are v2_03_03
-- CPP0X version and qualifier are v1_03_14 debug:e2
-- LIBSICPP version and qualifier are v2_2_10 debug:e2
-- CPPUNIT version and qualifier are v1_12_1 debug:e2
-- CLHEP version and qualifier are v2_1_2_4 debug:e2
-- CETLIB version and qualifier are v1_03_14 debug:e2
-- GCCXML version and qualifier are v0_9_20120702 gcc47
-- ART version and qualifier are v1_02_02 debug:e2
-- FHICL_CPP version and qualifier are v2_17_01 debug:e2
-- MESSAGEFACILITY version and qualifier are v1_10_15 debug:e2
-- ROOT version and qualifier are v5_34_01 debug:e2
```

```

-- Boost version and qualifier are v1_50_0 debug:e2
-- art_make: no library for /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/Lesson1
-- art_make: no library for /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/Lesson2
-- full_qualifiers: e2:debug e2-debug
-- CPACK_PACKAGE_NAME and CPACK_SYSTEM_NAME are gm2artexamples slf5-x86_64-gcc47-e2-debug
-- Configuring done
-- Generating done
-- Build files have been written to: /gm2/app/users/lyon/work-20121003/build

-----
INFO: Stage cmake successful.
-----

-----
INFO: gm2artexamples version v0_3.v0_3.v0_3 configured.
-----

-----
INFO: Stage build.
-----

INFO: current directory /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/Lesson1 is not in the current build tree
(/gm2/app/users/lyon/work-20121003/build). Executing stage build from $CETPKG_BUILD.

[ 5%] Built target gm2artexamples_DataObjects
[ 15%] Built target gm2artexamples_DataObjects_dict
[ 25%] Built target gm2artexamples_DataObjects_map
[ 35%] Built target gm2artexamples_HitAndTrackObjects_dict
[ 45%] Built target gm2artexamples_HitAndTrackObjects_map
[ 50%] Built target gm2artexamples_hitandtrackobjects
Scanning dependencies of target gm2artexamples_Lesson1_HelloWorld1_module
[ 55%] Building CXX object gm2artexamples/Lesson1/CMakeFiles/gm2artexamples_Lesson1_HelloWorld1_module.dir/HelloWorld1_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson1_HelloWorld1_module.so
[ 55%] Built target gm2artexamples_Lesson1_HelloWorld1_module
[ 60%] Built target gm2artexamples_Lesson1_HelloWorld2_module
[ 65%] Built target gm2artexamples_Lesson1_MyDatumReader_module
[ 70%] Built target gm2artexamples_Lesson1_ProduceMyLittleDatum_module
[ 75%] Built target gm2artexamples_Lesson2_makeHits_module
[ 80%] Built target gm2artexamples_Lesson2_readHits_module
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!MyLittleDatum_test.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!MyLittleDatum_test.d!fcl!MyLittleDatum_test.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!MyLittleDatum_test.d!fcl!messageDefaults.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!MyLittleDatum_r.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!MyLittleDatum_w.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!messageDefaults.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!simple_test.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!test_with_boost.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build!gm2artexamples!test!very_simple_test.sh.d
[ 85%] Built target gm2artexamples_test_MyLittleDatumAnalyzer_module
[ 90%] Built target gm2artexamples_test_MyLittleDatumProducer_module
[ 90%] Built target myLittleDatum_wr.sh
[ 95%] Built target simple_test
[100%] Built target test_with_boost
[100%] Built target very_simple_test.sh

real    0m11.698s
user    0m5.939s
sys      0m2.290s

-----
INFO: Stage build successful.
-----

```

This build was much faster than the full build from before.

Running it gives,

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature *)]$ gm2 -c hello1.fcl
%MSG-i MF_INIT_OK: gm2 04-Oct-2012 00:58:19 CDT JobSetup
Messagelogger initialization complete.
%MSG
Begin processing the 1st record. run: 1 subRun: 0 event: 1 at 04-Oct-2012 00:58:20 CDT
Hello, g-2. From analyze. run: 1 subRun: 0 event: 1
Begin processing the 2nd record. run: 1 subRun: 0 event: 2 at 04-Oct-2012 00:58:20 CDT
Hello, g-2. From analyze. run: 1 subRun: 0 event: 2

TrigReport ----- Event Summary -----
TrigReport Events total = 2 passed = 2 failed = 0

TrigReport ----- Modules in End-Path: path1 -----
TrigReport Trig Bit# Visited Passed Failed Error Name
TrigReport 0 0 2 2 0 0 hello

TimeReport ----- Time Summary ---[sec]----
TimeReport CPU = 0.009998 Real = 0.005523

Art has completed and will exit with status 0.
```

And there we have it!

3.1 Committing changed code

Let's ask git what has changed.

```
> git status
# On branch feature/myNewFeature
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   HelloWorld1_module.cc
#
no changes added to commit (use "git add" and/or "git commit -a")
```

There's our changed file. Let's commit it to **our** repository.

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature *)]$ git add HelloWorld1_module.cc
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature +)]$ git commit -m 'Changed Hello, world to Hello, g-2'
[feature/myNewFeature b4839e2] Changed Hello, world to Hello, g-2
1 file changed, 1 insertion(+), 1 deletion(-)
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ git status
# On branch feature/myNewFeature
nothing to commit (working directory clean)
```

Note that **ONLY** your git repository has been updated. No one else sees your change. If you delete your development area, no one will see these changes.

3.2 Finishing the feature branch and merging back into the develop branch

Let's say we're done with this feature and we want our changes to go into the develop branch. Once we do this, our changes will be seen by other people. The first thing we need to do is ensure that no one is surprised by us committing new code to develop. You should send e-mail to other people developing this application. Once you have done so, you need to do several steps.

You must make sure that no one else has updated the develop branch in the meantime. You first need to update your git repository with what is in Redmine. This does **not** happen automatically. Use the `git fetch` command.

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ git fetch
```

There is no output, which means there were no updates to develop you need to worry about. If there were changes, you would see something like,

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ git fetch
remote: Counting objects: 7, done.
remote: Compressing objects: 25% (1/4)
remote: Compressing objects: 50% (2/4)
remote: Compressing objects: 75% (3/4)    remote: Compressing objects: 100% (4/4)
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 3), reused 0 (delta 0)
Unpacking objects: 25% (1/4)
Unpacking objects: 50% (2/4)
Unpacking objects: 75% (3/4)    Unpacking objects: 100% (4/4)
Unpacking objects: 100% (4/4), done.
From ssh://cdcv.s.fnal.gov/cvs/projects/gm2artexamples
   caa8c62..d9d0505  develop    -> origin/develop
```

Again, this would look cleaner. Indeed the last line shows that there was a change to the develop branch on Redmine that we don't have in our repository. Let's see how far behind we are and update it. To do this, we need to temporarily change to the develop branch. First, we need to check what branch we're on.

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ git branch
develop
* feature/myNewFeature
master
```

We're on our feature branch. Is everything checked in?

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ git status
# On branch feature/myNewFeature
nothing to commit (working directory clean)
```

Yes, everything is checked in. We can now change to the develop branch. **IMPORTANT:** Had there been un-checked in changes, those changes would follow us when we switched to the develop branch! This is a feature, but in this case it would be confusing. Better to have everything checked in on our feature

Changing to the develop branch,

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ git checkout develop
Switched to branch 'develop'
Your branch is behind 'origin/develop' by 1 commit, and can be fast-forwarded.
```

Let's verify

```
[lyon@gm2gpvm01 Lesson1 (develop<)]$ git branch
* develop
  feature/myNewFeature
  master
```

We're on develop. Let's see again the status of our develop branch.

```
[lyon@gm2gpvm01 Lesson1 (develop<)]$ git status
# On branch develop
# Your branch is behind 'origin/develop' by 1 commit, and can be fast-forwarded.
```

```
#
nothing to commit (working directory clean)
```

So there has been one commit on Redmine (called `origin` by git) that we don't have (our `develop` is behind `origin/develop` by 1 commit). The message says we can “fast-forward”, which means that we can pull the commit without having to worry about a merge conflict. Let's see what's changed.

```
[lyon@gm2gpvm01 Lesson1 (develop=)]$ git diff origin/develop # Compare our branch with develop in Redmine (origin/develop)
diff --git a/Lesson1/HelloWorld2_module.cc b/Lesson1/HelloWorld2_module.cc
index c6ac975..c6ff29a 100644
--- a/Lesson1/HelloWorld2_module.cc
+++ b/Lesson1/HelloWorld2_module.cc
@@ -3,7 +3,7 @@
 // Another simple ART module that handles a configuration parameter and
 // prints a message for each event

-// See @HelloWorld1@ for more basic information.
+// See @HelloWorld1@ for more basic information

 // The skeleton of this code was generated with
 // <pre>artmod analyzer artex::HelloWorld2</pre>
@@ -109,4 +109,4 @@ void artex::HelloWorld2::endJob(){

 // Some boiler plate for the Art system. Note that the namespace is handled
 // differently here than in @HelloWorld1@
-DEFINE_ART_MODULE(artex::HelloWorld2)
+DEFINE_ART_MODULE(artex::HelloWorld2)
\ No newline at end of file
```

So it looks like a period was added to a comment (The `// See ... line`).

Ok – let's do the fast forward merge. The easy way to do this is to do a `git pull`. Because we are on `develop` and because `develop` tracks `origin/develop` (gm2d getRedmineGit set that up), `git pull` will automatically know what to do.

```
[lyon@gm2gpvm01 Lesson1 (develop<)]$ git pull
Updating caa8c62..d9d0505
Fast-forward
 Lesson1/HelloWorld2_module.cc | 4 +---
 1 file changed, 2 insertions(+), 2 deletions(-)
```

Ok – so now our `develop` branch is up to date. Let's finish our feature branching again. We change back to it with,

```
[lyon@gm2gpvm01 Lesson1 (develop=)]$ git checkout feature/myNewFeature
Switched to branch 'feature/myNewFeature'
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ git branch
 develop
* feature/myNewFeature
 master
```

Everything checked in?

```
[lyon@gm2gpvm01 Lesson1 (feature/myNewFeature)]$ git status
# On branch feature/myNewFeature
nothing to commit (working directory clean)
```

Ok, we use `git flow feature finish` to finish the feature branch.

```
[lyon@gm2gpvm01 Lesson1 (develop +[MERGING=)]$ git flow feature finish
Switched to branch 'develop'
[?1049h[?1h=[?25h[?25l".git/MERGE_MSG"
".git/MERGE_MSG" 7L, 277CMerge branch 'feature/myNewFeature' into develop

# Please enter a commit message to explain why this merge is necessary,
```

```
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
[?25l".git/MERGE_MSG"
".git/MERGE_MSG" 9L, 298C written
[?1l>[?25h[?1049lMerge made by the 'recursive' strategy.
 Lesson1/HelloWorld1_module.cc | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
Deleted branch feature/myNewFeature (was b4839e2).

Summary of actions:
- The feature branch 'feature/myNewFeature' was merged into 'develop'
- Feature branch 'feature/myNewFeature' has been removed
- You are now on branch 'develop'
```

Note that this will pop up vi (or whatever \$EDITOR is set to) for you to input a commit message. You should describe what is in this new feature. (The spurious characters in the output are due to vi).

So your develop branch now has your new feature. We should be on the develop branch now.

```
[lyon@gm2gpvm01 Lesson1 (develop>)]$ git branch
* develop
  master
```

BUT this branch is still local to you! You need to push it in order for others to see it. [PLEASE DO NOT ACTUALLY RUN THE FOLLOWING COMMAND AS IT WILL CHANGE THE REDMINE REPOSITORY]

```
[lyon@gm2gpvm01 Lesson1 (develop>)]$ git push origin develop
Counting objects: 12, done.
Delta compression using up to 2 threads.
Compressing objects: 14% (1/7)
Compressing objects: 28% (2/7)
Compressing objects: 42% (3/7)
Compressing objects: 57% (4/7)
Compressing objects: 71% (5/7)
Compressing objects: 85% (6/7)
Compressing objects: 100% (7/7)
Compressing objects: 100% (7/7), done.
Writing objects: 14% (1/7)
Writing objects: 28% (2/7)
Writing objects: 42% (3/7)
Writing objects: 57% (4/7)
Writing objects: 71% (5/7)
Writing objects: 85% (6/7)
Writing objects: 100% (7/7)
Writing objects: 100% (7/7), 692 bytes, done.
Total 7 (delta 5), reused 0 (delta 0)
To ssh://p-gm2artexamples@cdcv.s.fnl.gov/cvs/projects/gm2artexamples
 d9d0505..2b555c1 develop -> develop
```

And you are done.

4 Collaborative development on the same feature branch

There may be cases where more than one of you want to work on the same feature branch. This is in fact easy to do.

One of you has to first create the branch. Following the instructions above, you would do,

```
[lyon@gm2gpvm01 Lesson1 (develop=)]$ git flow feature start addTracking
Switched to a new branch 'feature/addTracking'
```

Summary of actions:

- A new branch 'feature/addTracking' was created, based on 'develop'
- You are now on branch 'feature/addTracking'

Now, start committing on your feature. When done, use:

```
git flow feature finish addTracking
```

You now have to “publish” your branch. This will push it to Redmine.

```
[lyon@gm2gpvm01 Lesson1 (feature/addTracking)]$ git flow feature publish addTracking
Total 0 (delta 0), reused 0 (delta 0)
To ssh://p-gm2artexamples@cdcv.s.fnal.gov/cvs/projects/gm2artexamples
 * [new branch]      feature/addTracking -> feature/addTracking
Already on 'feature/addTracking'
```

Summary of actions:

- A new remote branch 'feature/addTracking' was created
- The local branch 'feature/addTracking' was configured to track the remote branch
- You are now on branch 'feature/addTracking'

You can start developing. Your friend Bob would setup his environment, choosing the same g-2 release as you (but not necessarily the same build type). He would do `git getRedmineGit` on the product, putting him in the develop branch.

Bob (and you) can see the list of branches that are on Redmine with `git branch -a` (the `-a` means to list all branches, even remote ones).

```
[lyon@gm2gpvm01 build]$ git branch -a
develop
* feature/addTracking
master
remotes/origin/HEAD -> origin/master
remotes/origin/develop
remotes/origin/feature/BrendanSandBox
remotes/origin/feature/addTracking
remotes/origin/feature/initial
remotes/origin/feature/mwmcxx11
remotes/origin/master
```

Bob wants your feature/addTracking branch, so he would do.

```
git flow feature track addTracking
```

You both can develop. When you want Bob to see the code you’ve changed, you need to commit those changes (remember, that only changes **YOUR** repository). You can then “push” your changes to Redmine with.

```
[lyon@gm2gpvm01 Lesson1 (feature/addTracking=)]$ git push origin feature/addTracking
Counting objects: 7, done.
Delta compression using up to 2 threads.
Compressing objects: 25% (1/4)
Compressing objects: 50% (2/4)
Compressing objects: 75% (3/4)
Compressing objects: 100% (4/4)
Compressing objects: 100% (4/4), done.
Writing objects: 25% (1/4)
Writing objects: 50% (2/4)
Writing objects: 75% (3/4)
Writing objects: 100% (4/4)
Writing objects: 100% (4/4), 348 bytes, done.
Total 4 (delta 3), reused 0 (delta 0)
To ssh://p-gm2artexamples@cdcv.s.fnal.gov/cvs/projects/gm2artexamples
```

```
2b555c1..1053a18 feature/addTracking -> feature/addTracking
```

Since this is a bare `git` command, you need the full name of the branch (e.g. with the `feature/`).

Hopefully, you have notified Bob that your new code is up. Bob can do several things.

1. He can do a `git fetch` and then `git diff origin/addTracking` to see what is changed. If he likes it, he can do a `git pull`.
2. Bob can do a `git pull` immediately

Only one of you can do the `git flow feature finish`

5. Profile (optimized) builds

All of the examples above involve a debug build. This is typically the way you start with developing. We do have several debuggers you can use (the usual `gdb` and we also have `totalview`). Debuggers are beyond the scope of this tutorial. When you think things are working, you should switch to an optimized build. We'll actually use a **profile** build, which is the same as optimized, but has some hooks in it so that a profiler can determine function timings, etc. This has no impact on speed. So where you see "profile", read "optimized".

5.1 Setting up your profile environment

You must setup your development environment to a profile g-2 release. Since this is a drastic change to your environment, it is often best to log out and log back in again so that you start fresh. So do so now and continue here when you log back in.

cd to your development area

```
[lyon@gm2gpvm01 work-20121003]$ cd /gm2/app/users/lyon/work-20121003
```

Now, we must choose a profile g-2 build. Do what we did when we started a brand new development area above.

```
[lyon@gm2gpvm01 work-20121003]$ source /gm2/app/software/prod/g-2/setup
g-2 software

--> To list gm2 releases, type
ups list -aK+ gm2

--> To use the latest gm2 release, type
setup gm2 v201209 -q e2:debug

--> After the set up above, use the "gm2d" command to do development stuff
```

But now, choose a profile build. A profile build is not listed in the hint, but you can find it by listing all of the releases.

```
[lyon@gm2gpvm01 work-20121003]$ ups list -aK+ gm2
"gm2" "v201209" "Linux64bit+2.6-2.5" "debug:e2" ""
"gm2" "v201209" "Linux64bit+2.6-2.5" "e2:prof" ""
```

And there it is "e2:prof"

```
[lyon@gm2gpvm01 work-20121003]$ setup gm2 v201209 -q e2:prof
```

Now we have to **add** this development environment to our development area. We use the **-p** option to `gm2d newDev`, which says to add an environment to an already existing development area.

```
[lyon@gm2gpvm01 work-20121003]$ gm2d newDev -p
NOTICE: Just make products area
NOTICE: Created local products directory ./localProducts_v201209_e2-prof
NOTICE: Copied .upsfiles to ./localProducts_v201209_e2-prof
NOTICE: Created ./localProducts_v201209_e2-prof/setup

IMPORTANT: You must type
    source ./localProducts_v201209_e2-prof/setup
NOW and whenever you log in
```

So now we have a new directory called `localProducts_v201209_e2-prof` with a setup script. Source it. This is also the setup script to use when you log back in and want to continue in your profile environment.

```
[lyon@gm2gpvm01 work-20121003]$ source localProducts_v201209_e2-prof/setup
Executed setup gm2 v201209 -q e2:prof
```

5.2 Making a profile build

We need to set up a build directory. We could re-use `build`, but instead let's make a directory called `build-prof` so that we can compare the debug and non-debug executables (why not).

```
[lyon@gm2gpvm01 work-20121003]$ pwd
/gm2/app/users/lyon/work-20121003
[lyon@gm2gpvm01 work-20121003]$ mkdir build-prof
[lyon@gm2gpvm01 work-20121003]$ cd build-prof
```

Now we do the `setup_for_development` with the `-p` option for profile.

```
[lyon@gm2gpvm01 build-prof]$ source gm2d setup_for_development -p
--- Running gm2artexamples setup_for_development
The working build directory is /gm2/app/users/lyon/work-20121003/build-prof
The source code directory is /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples
----- check this block for errors -----
-----
/gm2/app/users/lyon/work-20121003/build-prof/lib has been added to LD_LIBRARY_PATH
/gm2/app/users/lyon/work-20121003/build-prof/bin has been added to PATH

CETPKG_SOURCE=/gm2/app/users/lyon/work-20121003/srcs/gm2artexamples
CETPKG_BUILD=/gm2/app/users/lyon/work-20121003/build-prof
CETPKG_NAME=gm2artexamples
CETPKG_VERSION=v0_3
CETPKG_QUAL=e2:prof
CETPKG_TYPE=Prof

Please use this cmake command:
cmake -DCMAKE_INSTALL_PREFIX=/install/path -DCMAKE_BUILD_TYPE=$CETPKG_TYPE $CETPKG_SOURCE
    Setting GM2ARTEEXAMPLES_DIR=/gm2/app/users/lyon/work-20121003/build-prof/gm2artexamples
-----
Define LD_LIBRARY_PATH to include /gm2/app/users/lyon/work-20121003/build-prof/lib
Define FHICL_FILE_PATH to include /gm2/app/users/lyon/work-20121003/build-prof/fcl
Summary:
  CETPKG_SOURCE   = /gm2/app/users/lyon/work-20121003/srcs
  CETPKG_BUILD    = /gm2/app/users/lyon/work-20121003/build-prof
  CETPKG_QUAL     = e2:prof
  CETPKG_TYPE     = CETPKG_TYPE
  CETPKG_INSTALL  = /gm2/app/users/lyon/work-20121003/localProducts_v201209_e2-prof
Run buildtool to build
```

And build with `gm2d build`

```
[lyon@gm2gpvm01 build-prof]$ gm2d build
INFO: CETPKG_TYPE = CETPKG_TYPE
```

```

-----
INFO: Stage cmake.
-----

-- The C compiler identification is GNU 4.7.1
-- The CXX compiler identification is GNU 4.7.1
-- Check for working C compiler: /gm2/app/software/prod/external/gcc/v4_7_1/Linux64bit+2.6-2.5/bin/gcc
-- Check for working C compiler: /gm2/app/software/prod/external/gcc/v4_7_1/Linux64bit+2.6-2.5/bin/gcc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /gm2/app/software/prod/external/gcc/v4_7_1/Linux64bit+2.6-2.5/bin/c++
-- Check for working CXX compiler: /gm2/app/software/prod/external/gcc/v4_7_1/Linux64bit+2.6-2.5/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Product is gm2artexamples v0_3 e2:prof
-- Module path is /gm2/app/software/prod/g-2/gm2softwaretools/v201209_01/Modules;/gm2/app/software/prod/external/cetbuildtools/v2_03_03/Modules
-- version parses to v0_3.v0_3.v0_3
-- project version is v0_3.v0_3.v0_3
-- Building for Linux slf5 x86_64
-- set_flavor_qual: flavorqual directory is gm2artexamples/v0_3/slf5.x86_64.e2.prof
-- Selected diagnostics option VIGILANT
-- cmake build type set to CETPKG_TYPE in directory gm2artexamples and below
-- compiler flags for directory gm2artexamples and below
--   C++ FLAGS:      -std=c++0x -Wno-deprecated-declarations
--   C   FLAGS:
-- CETBUILDTOOLS version and qualifier are v2_03_03
-- CPP0X version and qualifier are v1_03_14 e2:prof
-- LIBSIGCPP version and qualifier are v2_2_10 e2:prof
-- CPPUNIT version and qualifier are v1_12_1 e2:prof
-- CLHEP version and qualifier are v2_1_2_4 e2:prof
-- CETLIB version and qualifier are v1_03_14 e2:prof
-- GCCXML version and qualifier are v0_9_20120702 gcc47
-- ART version and qualifier are v1_02_02 e2:prof
-- PHICLCP version and qualifier are v2_17_01 e2:prof
-- MESSAGEFACILITY version and qualifier are v1_10_15 e2:prof
-- ROOT version and qualifier are v5_34_01 e2:prof
-- Boost version and qualifier are v1_50_0 e2:prof
-- Boost version: 1.50.0
-- Found the following Boost libraries:
--   filesystem
--   program_options
--   regex
--   thread
--   unit_test_framework
-- art_make: no library for /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/Lesson1
-- art_make: no library for /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/Lesson2
-- full_qualifiers: e2:prof e2-prof
-- CPACK_PACKAGE_NAME and CPACK_SYSTEM_NAME are gm2artexamples slf5-x86_64-gcc47-e2-prof
-- Configuring done
-- Generating done
-- Build files have been written to: /gm2/app/users/lyon/work-20121003/build-prof

-----
INFO: Stage cmake successful.
-----

-----
INFO: gm2artexamples version v0_3.v0_3.v0_3 configured.
-----

-----
INFO: Stage build.
-----

Scanning dependencies of target gm2artexamples_DataObjects

```

```
[ 5%] Building CXX object gm2artexamples/DataObjects/CMakeFiles/gm2artexamples_DataObjects.dir/MyLittleDatum.cc.o
Linking CXX shared library ../lib/libgm2artexamples_DataObjects.so
[ 5%] Built target gm2artexamples_DataObjects
[ 10%] Generating gm2artexamples_DataObjects_dict.cpp, gm2artexamples_DataObjects_map.cpp
---> genreflex: INFO: Using gccxml from /gm2/app/software/prod/external/gccxml/v0_9_20120702/Linux64bit+2.6-2.5-gcc47/bin/gccxml
---> genreflex: INFO: Parsing file /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/DataObjects/classes.h with GCC_XML OK
---> genreflex: INFO: Generating Reflex Dictionary
class artex::MyLittleDatum
class std::vector<artex::MyLittleDatum>
class art::Wrapper<std::vector<artex::MyLittleDatum> >
Scanning dependencies of target gm2artexamples_DataObjects_dict
[ 15%] Building CXX object gm2artexamples/DataObjects/CMakeFiles/gm2artexamples_DataObjects_dict.dir/gm2artexamples_DataObjects_dict.cpp.o
Linking CXX shared library ../lib/libgm2artexamples_DataObjects_dict.so
[ 15%] Built target gm2artexamples_DataObjects_dict
Scanning dependencies of target gm2artexamples_DataObjects_map
[ 20%] Building CXX object gm2artexamples/DataObjects/CMakeFiles/gm2artexamples_DataObjects_map.dir/gm2artexamples_DataObjects_map.cpp.o
Linking CXX shared library ../lib/libgm2artexamples_DataObjects_map.so
[ 25%] Built target gm2artexamples_DataObjects_map
[ 30%] Generating gm2artexamples_HitAndTrackObjects_dict.cpp, gm2artexamples_HitAndTrackObjects_map.cpp
---> genreflex: INFO: Using gccxml from /gm2/app/software/prod/external/gccxml/v0_9_20120702/Linux64bit+2.6-2.5-gcc47/bin/gccxml
---> genreflex: INFO: Parsing file /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples/HitAndTrackObjects/classes.h with GCC_XML OK
---> genreflex: INFO: Generating Reflex Dictionary
class artex::Hit
class std::vector<artex::Hit>
class CLHEP::Hep3Vector
class art::Wrapper<std::vector<artex::Hit> >
Scanning dependencies of target gm2artexamples_HitAndTrackObjects_dict
[ 35%] Building CXX object gm2artexamples/HitAndTrackObjects/CMakeFiles/gm2artexamples_HitAndTrackObjects_dict.dir/gm2artexamples_HitAndTrackObjects_dict.cpp.o
Linking CXX shared library ../lib/libgm2artexamples_HitAndTrackObjects_dict.so
[ 35%] Built target gm2artexamples_HitAndTrackObjects_dict
Scanning dependencies of target gm2artexamples_HitAndTrackObjects_map
[ 40%] Building CXX object gm2artexamples/HitAndTrackObjects/CMakeFiles/gm2artexamples_HitAndTrackObjects_map.dir/gm2artexamples_HitAndTrackObjects_map.cpp.o
Linking CXX shared library ../lib/libgm2artexamples_HitAndTrackObjects_map.so
[ 45%] Built target gm2artexamples_HitAndTrackObjects_map
Scanning dependencies of target gm2artexamples_hitandtrackobjects
[ 50%] Building CXX object gm2artexamples/HitAndTrackObjects/CMakeFiles/gm2artexamples_hitandtrackobjects.dir/Hit.cpp.o
Linking CXX shared library ../lib/libgm2artexamples_hitandtrackobjects.so
[ 50%] Built target gm2artexamples_hitandtrackobjects
Scanning dependencies of target gm2artexamples_Lesson1>HelloWorld1_module
[ 55%] Building CXX object gm2artexamples/Lesson1/CMakeFiles/gm2artexamples_Lesson1>HelloWorld1_module.dir/HelloWorld1_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson1>HelloWorld1_module.so
[ 55%] Built target gm2artexamples_Lesson1>HelloWorld1_module
Scanning dependencies of target gm2artexamples_Lesson1>HelloWorld2_module
[ 60%] Building CXX object gm2artexamples/Lesson1/CMakeFiles/gm2artexamples_Lesson1>HelloWorld2_module.dir/HelloWorld2_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson1>HelloWorld2_module.so
[ 60%] Built target gm2artexamples_Lesson1>HelloWorld2_module
Scanning dependencies of target gm2artexamples_Lesson1_MyDatumReader_module
[ 65%] Building CXX object gm2artexamples/Lesson1/CMakeFiles/gm2artexamples_Lesson1_MyDatumReader_module.dir/MyDatumReader_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson1_MyDatumReader_module.so
[ 65%] Built target gm2artexamples_Lesson1_MyDatumReader_module
Scanning dependencies of target gm2artexamples_Lesson1_ProduceMyLittleDatum_module
[ 70%] Building CXX object gm2artexamples/Lesson1/CMakeFiles/gm2artexamples_Lesson1_ProduceMyLittleDatum_module.dir/ProduceMyLittleDatum_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson1_ProduceMyLittleDatum_module.so
[ 70%] Built target gm2artexamples_Lesson1_ProduceMyLittleDatum_module
Scanning dependencies of target gm2artexamples_Lesson2_makeHits_module
[ 75%] Building CXX object gm2artexamples/Lesson2/CMakeFiles/gm2artexamples_Lesson2_makeHits_module.dir/makeHits_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson2_makeHits_module.so
[ 75%] Built target gm2artexamples_Lesson2_makeHits_module
Scanning dependencies of target gm2artexamples_Lesson2_readHits_module
[ 80%] Building CXX object gm2artexamples/Lesson2/CMakeFiles/gm2artexamples_Lesson2_readHits_module.dir/readHits_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_Lesson2_readHits_module.so
[ 80%] Built target gm2artexamples_Lesson2_readHits_module
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!MyLittleDatum_test.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!MyLittleDatum_test.d
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!MyLittleDatum_test.d!fcl!MyLittleDatum_test.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!MyLittleDatum_test.d!fcl!MyLittleDatum_test.fcl
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!MyLittleDatum_test.d!fcl!messageDefaults.fcl
```

```
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!MyLittleDatum_test.d!fcl!messageDefaults.fcl
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!myLittleDatum_wr.sh.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!myLittleDatum_wr.sh.d
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!myLittleDatum_r.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!MyLittleDatum_r.fcl
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!myLittleDatum_w.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!MyLittleDatum_w.fcl
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!messageDefaults.fcl
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!myLittleDatum_wr.sh.d!fcl!messageDefaults.fcl
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!simple_test.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!simple_test.d
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!test_with_boost.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!test_with_boost.d
Scanning dependencies of target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!very_simple_test.sh.d
[ 80%] Built target !gm2!app!users!lyon!work-20121003!build-prof!gm2artexamples!test!very_simple_test.sh.d
Scanning dependencies of target gm2artexamples_test_MyLittleDatumAnalyzer_module
[ 85%] Building CXX object gm2artexamples/test/CMakeFiles/gm2artexamples_test_MyLittleDatumAnalyzer_module.dir/MyLittleDatumAnalyzer_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_test_MyLittleDatumAnalyzer_module.so
[ 85%] Built target gm2artexamples_test_MyLittleDatumAnalyzer_module
Scanning dependencies of target gm2artexamples_test_MyLittleDatumProducer_module
[ 90%] Building CXX object gm2artexamples/test/CMakeFiles/gm2artexamples_test_MyLittleDatumProducer_module.dir/MyLittleDatumProducer_module.cc.o
Linking CXX shared library ../lib/libgm2artexamples_test_MyLittleDatumProducer_module.so
[ 90%] Built target gm2artexamples_test_MyLittleDatumProducer_module
Scanning dependencies of target myLittleDatum_wr.sh
[ 90%] Built target myLittleDatum_wr.sh
Scanning dependencies of target simple_test
[ 95%] Building CXX object gm2artexamples/test/CMakeFiles/simple_test.dir/simple_test.cc.o
Linking CXX executable ../bin/simple_test
[ 95%] Built target simple_test
Scanning dependencies of target test_with_boost
[100%] Building CXX object gm2artexamples/test/CMakeFiles/test_with_boost.dir/test_with_boost.cc.o
Linking CXX executable ../bin/test_with_boost
[100%] Built target test_with_boost
Scanning dependencies of target very_simple_test.sh
[100%] Built target very_simple_test.sh

real    4m36.783s
user    1m27.627s
sys      0m14.158s
```

```
-----
INFO: Stage build successful.
-----
```

And let's try it

```
[lyon@gm2gpvm01 build-prof]$ gm2 -c hello1.fcl
%MSG-i MF_INIT_OK: gm2 04-Oct-2012 13:07:36 CDT JobSetup
MessageLogger initialization complete.
%MSG
Begin processing the 1st record. run: 1 subRun: 0 event: 1 at 04-Oct-2012 13:07:37 CDT
Hello, g-2. From analyze. run: 1 subRun: 0 event: 1
Begin processing the 2nd record. run: 1 subRun: 0 event: 2 at 04-Oct-2012 13:07:37 CDT
Hello, g-2. From analyze. run: 1 subRun: 0 event: 2

TrigReport ----- Event Summary -----
TrigReport Events total = 2 passed = 2 failed = 0

TrigReport ----- Modules in End-Path: path1 -----
TrigReport Trig Bit# Visited Passed Failed Error Name
TrigReport 0 0 2 2 0 0 hello

TimeReport ----- Time Summary ---[sec]----
TimeReport CPU = 0.009999 Real = 0.001420

Art has completed and will exit with status 0.
```

5.3 Using the same directory for profile and debug builds

You can use the same directory for profile and debug build. You can't have both at the same time, but you can switch from one to the other. But every time you do so, you must rebuild everything (so having two directories is better). But here's how to do it.

We already have the `build` directory which holds a debug build. Let's change it to a profile build. We have to remove everything in `build`. To make this easy, you can use the `gm2d zapBuild` command. This command ensures that you are in a build directory (a directory that starts with the word `build`).

```
[lyon@gm2gpvm01 build-prof]$ pwd
/gm2/app/users/lyon/work-20121003/build-prof
[lyon@gm2gpvm01 build]$ cd ../build
[lyon@gm2gpvm01 build]$ pwd
/gm2/app/users/lyon/work-20121003/build
[lyon@gm2gpvm01 build]$ gm2d zapBuild
Removing everything in your build directory
You must now run the following:
    source gm2d s <OPTIONS>    # gm2d s is short for gm2d setup_for_development
[lyon@gm2gpvm01 build]$ ls
```

The directory is indeed empty! So now, as the instructions state, we have to run `setup_for_development` again. We need to make sure we've already chosen a profile release of g-2. Let's check that.

```
[lyon@gm2gpvm01 build]$ echo $GM2_FQ_DIR
/gm2/app/software/prod/g-2/gm2/v201209/slf5.x86_64.e2.prof
```

Yep – it's a profile release. Had it been a debug release, you could have run `source ../localProducts_v201209_e2-prof/setup` to get to the right release. Let's run `setup_for_development`. Note the `-p`.

```
[lyon@gm2gpvm01 build]$ source gm2d s -p
--- Running gm2artexamples setup_for_development
The working build directory is /gm2/app/users/lyon/work-20121003/build
The source code directory is /gm2/app/users/lyon/work-20121003/srcs/gm2artexamples
----- check this block for errors -----
-----
/gm2/app/users/lyon/work-20121003/build/lib has been added to LD_LIBRARY_PATH
/gm2/app/users/lyon/work-20121003/build/bin has been added to PATH

CETPKG_SOURCE=/gm2/app/users/lyon/work-20121003/srcs/gm2artexamples
CETPKG_BUILD=/gm2/app/users/lyon/work-20121003/build
CETPKG_NAME=gm2artexamples
CETPKG_VERSION=v0_3
CETPKG_QUAL=e2:prof
CETPKG_TYPE=Prof

Please use this cmake command:
cmake -DCMAKE_INSTALL_PREFIX=/install/path -DCMAKE_BUILD_TYPE=$CETPKG_TYPE $CETPKG_SOURCE
    Setting GM2ARTEEXAMPLES_DIR=/gm2/app/users/lyon/work-20121003/build/gm2artexamples
-----
Define LD_LIBRARY_PATH to include /gm2/app/users/lyon/work-20121003/build/lib
Define FHICL_FILE_PATH to include /gm2/app/users/lyon/work-20121003/build/fcl
Summary:
    CETPKG_SOURCE   = /gm2/app/users/lyon/work-20121003/srcs
    CETPKG_BUILD    = /gm2/app/users/lyon/work-20121003/build
    CETPKG_QUAL     = e2:prof
    CETPKG_TYPE     = CETPKG_TYPE
    CETPKG_INSTALL  = /gm2/app/users/lyon/work-20121003/localProducts_v201209_e2-prof
Run buildtool to build
```

And build the usual way...

To switch back to debug, follow the steps here but choose the debug release(`source localProducts_v201209_e2-debug/setup`), do `gm2d zapBuild` again, and `setup_for_development -d`.

